Technical Note: Detecting and Enforcing Structure via Commutation with Hadamard Involutions

1. Intuition: systems that preserve a hidden split

Imagine you have a way to split any signal into two perfectly non-interacting parts. Now consider a system (a channel, a transform, a neural layer):

- If the system **respects that split**, the two parts remain separate
- If it **doesn't**, they start to mix

That gives you a powerful idea:

> Instead of measuring error directly, measure whether the system preserves a known structural split.

For Walsh–Hadamard transforms, this becomes extremely clean—and even computable in closed form.

---

2. The core object: an involution

Let H be a matrix such that:

$H^2 = I$

The normalized Walsh–Hadamard transform is a canonical example.

This implies:

- eigenvalues are +1 and -1
- space splits into two orthogonal subspaces

Define projectors:

$P_+ = \frac{1}{2}(I+H), \quad P_-=\frac{1}{2}(I-H)$

These give a decomposition:

$x = P_+x + P_-x$

with:

- orthogonality
- perfect reconstruction
- zero cross-interaction in the ideal case

---

3. The key condition: AH = HA

Now consider a (system/operator) matrix A.

The condition:

$AH = HA$

means:

> **A preserves the +1 and −1 eigenspaces of H**

Equivalently:

- A does not move energy between the two subspaces
- the decomposition remains “clean” after applying A

---

4. What happens if A does not commute?

If:

$AH \neq HA$

then:

- A mixes the two subspaces
- components that should remain separate begin to interact

Which shows up as **leakage** between the 2 subspaces.

---

5. Exact decomposition of any system

Any matrix A can be split into:

- a **structure-preserving part** (commutes with H)
- a **structure-violating part** (does not)

Closed-form:

$A_{good} = ½(A+HAH)$

$A_{bad} = ½(A−HAH)$

Properties:

- $A_{good}H = HA_{good}$
- $A_{bad}H = −HA_{bad}$

So:

$A = A_{good} + A_{bad}$

---

6. Interpretation

- A_{good} : preserves the decomposition
- A_{bad} : causes mixing between subspaces

So:

> Measuring how large A_{bad} is tells you how much the system violates the structure.

---

7. Projection: enforcing the structure

You can *force* a system to satisfy AH = HA by projecting:

$A \mapsto \frac{1}{2}(A+HAH)$

This:

- removes all cross-subspace mixing
- is exact (no iteration needed)
- gives the closest commuting matrix (in a natural sense)

---

8. Geometric picture

Think of the space as split into two orthogonal “planes”:

- V_+ : eigenvalue (+1)
- V_- : eigenvalue (-1)

Then:

- commuting A: acts **within each plane only**
- non-commuting A: moves vectors between planes

Leakage = how much “cross-plane motion” occurs.

---

9. Why this is useful

This gives a new way to analyze systems:

Instead of:

- estimating noise directly
- modeling the full system

You can:

- check whether a known structural symmetry is preserved
- measure violations of that symmetry

---

10. Applications (conceptual)

Signal processing

- detect channel distortion via subspace mixing
- separate “structured” vs “unstructured” effects

Synchronization

- misalignment breaks commutation
- restoring commutation aligns the system

System diagnostics

- quantify deviation from ideal structure
- distinguish mixing from pure noise

Optimization / learning

- enforce constraints via projection
- penalize non-commuting components

---

11. Technical summary

- $H_2 = I \Rightarrow$ eigenspace split
- $AH = HA \Leftrightarrow$ no cross-subspace mixing
- decomposition:
    - commuting: ½(A + HAH))
    - non-commuting: ½(A - HAH))
- projection operator:
    - P(A) = ½(A+HAH)
    - idempotent (one-step)
- leakage corresponds to:
    - the -1 eigenspace of $A \mapsto HAH$

---

12. Brief note on randomization (optional extension)

Replacing H with:

H' = DHD

(where D has ±1 on the diagonal)

- gives another involution
- defines a different eigenspace split

Using multiple such transforms lets you:

- probe different structural alignments
- avoid dependence on a single fixed basis

---

13. Big picture

The condition:

AH = HA

is more than algebra—it defines a **structural symmetry** of the system.

This leads to a general principle:

> Choose a simple symmetry (like an involution),
> and measure how much a system fails to respect it.

Because for involutions, this failure:

- is exactly decomposable
- directly measurable
- and removable in closed form